Algorithm for efficient symbolic analysis of large analogue circuits

P. Wambacq, F.V. Fernández, G. Gielen, W. Sansen and A. Rodríguez-Vázquez

Indexing terms: Circuit analysis computing, Analogue circuits

An algorithm is presented that generates simplified symbolic expressions for the small-signal characteristics of large analogue circuits. The expressions are approximated while they are computed, so that only the most significant terms are generated which remain in the final expression. This principle leads to dramatic savings in CPU time and memory compared to existing techniques, significantly increasing the maximum size of circuits that can be analysed. By taking into account a range for the value of a circuit parameter rather than one single number the generated symbolic expressions are also generally valid.

Introduction: In recent years, several symbolic simulators, such as ISAAC [1] and ASAP [2], have been developed that can generate simplified symbolic expressions of any AC characteristic of an analogue circuit in the form of a rational function of the frequency variable s and of the symbolic circuit parameters. The approximation is based on the numerical values of the circuit parameters. These programs can be used both interactively to generate interpretable expressions, and in design automation systems for analogue integrated circuits. Their most important drawback is the limited size of circuits they can handle.

The program presented here no longer generates a complete symbolic expression prior to approximation, in contrast to conventional symbolic analysers such as ISAAC and ASAP. Instead, only the necessary terms are directly generated with a 'simplification during generation' procedure. In this way, larger circuits can be analysed than with conventional symbolic analysis [1–3].

General concept: The method described here uses an algorithm for the generation of terms of a network function in decreasing order of magnitude. The generation mechanism stops when the deviation between the numerical value of the sum of generated terms for each power of s and the exact total numerical value of the corresponding coefficient is below a predefined error. Hence, the numerical value of each coefficient must be computed in advance. This is performed with the polynominal interpolation method [3].

For the approximation of symbolic expressions, normally fixed numerical values are used for the circuit parameters. Hence, an approximated symbolic expression might become less accurate when it is used for different values of the circuit parameters. In the new algorithm, ranges are assigned to the circuit parameter values. These ranges are used both in the error control and in the calculation of the total numerical value of the coefficient of every power of s. In this way, the validity range of the generated symbolic expressions is extended.

Generation of symbolic terms: The basic technique used to generate symbolic terms of a network function is the undirected tree enumeration method. In this approach, a valid term corresponds to a common spanning tree of the voltage graph and the current graph which are directly constructed from the circuit topology. The number of trees increases exponentially with the circuit size. Because we are interested only in the dominant terms, the new algorithm enumerates spanning trees in the voltage graph in decreasing order. For every spanning tree, a check is made to determine whether the corresponding branches in the current graph constitute a spanning tree as well. If so, a valid term is found and its sign is determined.

This technique is performed for every power of the frequency variable s in both the numerator and denominator of the network function. For a nonzero power of s, say the kth power, spanning trees in decreasing order must be generated containing exactly k capacitance branches. This can be formulated as the following graph problem: given a graph with n nodes and with red and blue weighted branches, enumerate in decreasing order the spanning trees that contain exactly k blue branches and $n - k - 1$ red branches, in which k can have a value between zero and $n - 1$.

This problem is solved using an extension of the algorithm for the enumeration of spanning trees in decreasing order in a one-coloured graph [4]. The method repetitively uses a reference tree from which trees with smaller weight are deduced. Initially, the reference tree is the maximum spanning tree. In a one-coloured graph the remaining trees, which have a smaller weight, are derived from this reference tree using so-called T-exchanges [4]: for a given spanning tree T that contains a branch b_e with weight $w(b_e)$, a T-exchange $[b_e, b_f]$ is the replacement of branch b_e by branch b_f ($b_f \notin T$) with weight $w(b_f)$, such that $T - b_e \cup b_f$ is also a spanning tree. The weight of the T-exchange is $-w(b_e) + w(b_f)$. If T is the maximum spanning tree, then the next spanning tree is found by looking for the T-exchange with the least negative weight. To prevent spanning trees from being generated twice, the spanning trees that still have to be generated are partitioned into two disjoint sets: a set A containing all trees that include branch b_e, and a set B containing all trees that do not include b_e. The maximum spanning tree in A is derived from T by finding the least negative exchange from T, while the maximum spanning tree in B is derived from $T - b_e \cup b_f$. The least negative T-exchange among all partitions provides the next largest spanning tree. The partition that contains this tree is in turn divided into two partitions, and so on.

For the extension to a two-coloured graph, again a reference tree is used, which initially is the maximum spanning tree. The calculation of the maximum spanning tree in a two-coloured graph can make use of standard techniques for one-coloured graphs [5] applying suitable scale factors. The remaining trees can be derived from the reference tree using three kinds of T-exchange: a single T-exchange of two red branches, like the exchange described above, a single T-exchange of two blue branches, and a double T-exchange, consisting of a T-exchange $[b_e, r_f]$ of a blue branch b_e into a red one r_f together with a T-exchange $[r_e, b_f]$ of a red branch r_e into a blue one b_f. The weight of this double exchange is $-w(b_e) + w(r_f) - w(r_e) + w(b_f)$. When the exchange with the least negative weight is found to be a single T-exchange, then the corresponding partition is divided in two, just as in the one-coloured case. If the least negative exchange is double, then four new partitions are created that cover all spanning trees that still have to be generated:

(i) trees that contain both b_e and r_e

(ii) trees that contain b_e and do not contain r_e

(iii) trees that contain r_e and do not contain b_e

(iv) trees that contain neither b_e nor r_e

For each of these four partitions, single and double T-exchanges are updated and the least negative exchange provides the maximum spanning tree in that partition.

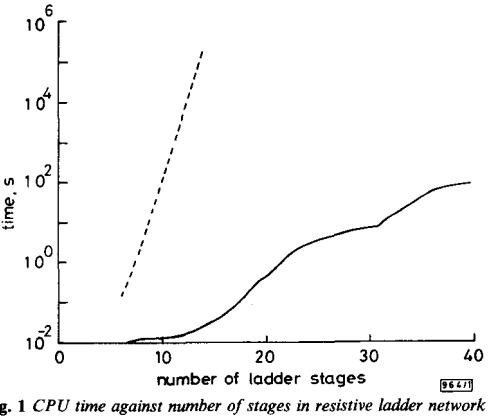

Fig. 1 *CPU time against number of stages in resistive ladder network*

– – – – ASAP
——— new approach

Examples: As a first example, the transfer function of a resistive ladder network is computed symbolically with an error of 25%. The number of terms of the exact expression for the denominator increases exponentially with the number of stages [3]. In Fig. 1, the CPU time is shown as a function of the number of ladder stages for both conventional symbolic analysis and the new approach. The dramatic increase in CPU time with the number of stages for conventional symbolic analysis is due to the necessity of generating the exact expression before the approximation phase.

Clearly, the new technique can generate an approximate expression in a CPU time that is up to several orders of magnitude smaller than with conventional analysis. Moreover, the new technique far exceeds the circuit size limits of a conventional symbolic analyser.

Another example is the symbolic computation of a μA741 operational amplifier. This circuit, containing 23 nodes, 22 transistors and 13 resistors, is much too complex to analyse with conventional symbolic analysis. The new approach computes the amplifier's transfer function with an error of 0.1% in 38s on a SUN SPARC 10 workstation, yielding an expression 110 terms long.

Conclusions: A new algorithm has been presented for generating the dominant terms of a network function in symbolic form. The method is based on the tree enumeration method. The algorithm can analyse large anlogue circuits which are too complex for previously reported symbolic analysers.

28 April 1994
Electronics Letters Online No: 19940788

P. Wambacq, F. Fernández, G. Gielen and W. Sansen (*Katholieke Universiteit Leuven, Dep. Elektrotechniek, ESAT-MICAS, Kardiaal Mercierlaan 94, B-3001 Heverlee, Belgium*)

A. Rodríguez-Vázquez (*Dept. of Analog Circuit Design, Centro Nacional de Microelectrónica, Edif. CICA, Avda. Reina Mercedes s/n, E-41012 Sevilla, Spain*)

V Fernández: Also with Dept. of Analog Circuit Design, Centro Nacional de Microelectrónica, Spain

References

1 GIELEN, G., WALSCHARTS, H., and SANSEN, W.: 'ISAAC: a symbolic simulator for analog integrated circuits', *IEEE J. Solid State Circuits*, 1989, **SC-24**, (6), pp. 1587–1597
2 FERNÁNDEZ, F.V., RODRÍGUEZ-VÁZQUEZ, A., and HUERTAS, J.L.: 'Interactive sc modeling and characterization of analog circuits via symbolic analysis' in 'Analog integrated circuits and signal processing, Vol. 1' (Kluwer, November 1991), pp. 183–208
3 LIN, P.-M.: 'Symbolic network analysis' (Elsevier, 1991)
4 GABOW, H.: 'Two algorithms for generating weighted spanning trees in order', *SIAM J. Comput.*, 1977, **6**, (1), pp. 139–150
5 CHERITON, D., and TARJAN, R.: 'Finding minimum spanning trees', *SIAM J. Comput.*, 1976, **5**, (4), pp. 725–742

Calculation of power diode reverse-recovery time for SPICE simulations

A.G.M. Strollo

Indexing terms: Power electronics, Diodes, Semiconductor device modelling, SPICE

A new model for the calculation of power diode reverse-recovery time is described. The model takes into account the effect of emitter recombination and is easily incorporated into the PSPICE simulator. Comparisons between the proposed model and numerical simulation results are presented.

Introduction: The *pin* diode is a key component in power electronics. An accurate modelling of its reverse-recovery behaviour is very important for power circuit simulation, as this process can lead to large overvoltage or overcurrent stresses in other components, particularly switches. Unfortunately, the standard diode model used in SPICE [1], based on integral charge-control equations for the low-injection regime, is inappropriate for analysing power *pin* structures. Consequently, a number of new approaches have been recently proposed for power diode modelling [2–6].

In a recent paper [7] a model for the *pin* diode reverse-recovery time was presented. The model [7] is based on the assumption of negligible recombination in the end regions and calculates the reverse recovery time by means of a convolution integral. In this Letter we propose an extension to the model of [7]. Our approach takes into account the emitter recombination effect, which strongly influences the power diode recovery time [8], and is easily incorporated into the PSPICE simulator [9] as a subcircuit. Moreover, the calculation of the convolution integral is avoided and a fast evaluation of the reverse-recovery time is obtained.

Proposed model: The carrier distribution in the base region of a power diode is governed by the ambipolar diffusion equation

$$\frac{\partial^2 p}{\partial x^2} = \frac{p}{L_a^2} + \frac{1}{D_a}\frac{\partial p}{\partial t} \tag{1}$$

where D_a is the ambipolar diffusion coefficient and L_a is the ambipolar diffusion length. The boundary conditions for eqn. 1 are given by

$$\frac{\partial p}{\partial x}(0,t) = -\frac{b}{1+b}\frac{J}{qD_a} + \frac{h_p}{D_a}p(0)^2 \tag{2}$$

$$\frac{\partial p}{\partial x}(w,t) = \frac{1}{1+b}\frac{J}{qD_a} - \frac{h_n}{D_a}p(w)^2 \tag{3}$$

where J is the diode current density, $b = \mu_n/\mu_p$ and the coefficients h_n and h_p account for emitter recombination [8]. The origin of the x-axis is assumed at the P^+i junction; the base width is indicated by w.

We consider the Laplace transform with respect to time of eqn. 1; this yields an ordinary differential equation for $P(x, s)$ (capital letters will be used in the following to indicate L-transformed quantities). By solving this equation we obtain

$$-\frac{dP}{dx}(0,s) = P(0,s)Y_p(s) + [P(0,s) - P(w,s)]/Z_m(s) \tag{4}$$

$$\frac{dP}{dx}(w,s) = P(w,s)Y_p(s) + [P(w,s) - P(0,s)]/Z_m(s) \tag{5}$$

with

$$Z_m = w\frac{\sinh(z)}{z} \qquad Y_p = \frac{1}{w}\frac{z[\cosh(z)-1]}{\sinh(z)} \tag{6}$$

where $z = (w/L_a)\sqrt{(1 + s\tau_a)}$ and τ_a is the high-level lifetime. Eqns. 4 and 5 can be interpreted [10] as the equations of a two-port network (see Fig. 1) in which node voltages correspond to carrier concentration while input and output currents correspond to the x-derivative of carrier concentration. The nonlinear boundary conditions (eqns. 2 and 3) are represented by two nonlinear current generators, G_L and G_R, controlled by J, $p(0)$ and $p(w)$.

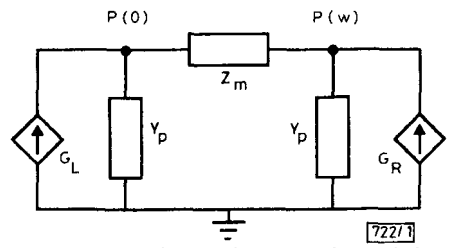

Fig. 1 *Two-port network describing base region of power diode*

To obtain a lumped equivalent circuit model, functions Z_m and Y_p are approximated with rational expressions. Function Z_m is approximated with the first two terms of its Taylor series [10]: $Z_m \simeq w(1 + z^2/6)$. In this way the impedance Z_m is represented by the series of a resistance $R_m = w(1 + k/6)$ and of an inductance $L_m = wk\tau_a/6$, where $k = (w/L_a)^2$. Function Y_p is approximated by using a rational function, as follows: $Y_p \simeq (1/w)(a_2z^2 + a_4z^4)/(1 + b_2z^2)$. The coefficients a_i, b_i are obtained by using the Padé approxima-